1. A method of transferring data via a communication session between a client application and a server application, the method comprising:

assigning an identifier to the communication session;

creating at least one queue associated with the communication session; and

storing data passed between the client application and the server application in the at least one queue, the data being stored using the identifier;

wherein the client application and the server application run local protocols, and the data is passed between the client application and the server application via an intermediary protocol.

2. The method of claim 1, further comprising:

creating a socket interface to at least one of the client application and the server application, the data being transmitted through the socket interface.

3. The method of claim 1, wherein the client application and the server application are on networks that run local protocols, and the method further comprises:

converting between the local protocols and the intermediary protocol when passing the data.

4. The method of claim 3, wherein the local protocol comprises at least one of TCP/IP and a serial protocol, the serial protocol comprising one of RS232 and RS485.

5. The method of claim 3 wherein the intermediary protocol comprises HTTP.

6. The method of claim 1, wherein the identifier is associated with the at least one queue.

7. The method of claim 1, wherein the method is performed by a server, and further comprises:

performing load balancing to select the server from among plural servers.

8. The method of claim 1, wherein the identifier is invalidated when the communication session terminates.

9. The method of claim 1, wherein the communication

session comprises a telnet session.

10. The method of claim 1, wherein the communication session is effected via a Web site.

11. The method of claim 1, further comprising maintaining a session record, the session record including an identity of a user initiating the session.

12. A system for transferring data via a communication session between a client application and a server application, the client application running on a first network and the server application running on a second network, the system comprising:

a proxy having a socket to the client application, the proxy converting data between a local protocol run on the first network to a non-local protocol;

an agent that creates a socket to the server application, the agent converting data between a local protocol run on the second network and the non-local protocol; and

a server in communication with the proxy and the agent, the server containing a message queue dedicated to the communication session, the message queue for storing data transmitted during the communication session.

13. The system of claim 12, wherein the proxy polls the server for data for the client application.

14. The system of claim 13, wherein, when data is present for the client application, the proxy retrieves the data from the message queue and passes the data to the client application.

15. The system of claim 12, wherein the agent polls the server for data for the server application.

16. The system of claim 15, wherein, when data is present for the client application, the agent retrieves the data from the message queue and passes the data to the server application.

17. A machine-readable medium that stores instructions for use in transferring data via a

communication session between a client application and a server application, the instructions causing a machine to:

assign an identifier to the communication session;

create at least one queue associated with the communication session; and

store data passed between the client application and the server application in the at least one queue, the data being stored using the identifier;

wherein the client application and the server application run local protocols, and the data is passed between the client application and the server application via an intermediary protocol.

18. The machine-readable medium of claim 17, wherein the intermediary protocol is different from the local protocols.

19. The method of claim 1, wherein the intermediary protocol is different from the local protocols.

20. The method of claim 1, wherein the intermediary protocol is a same protocol as the local protocols.